

```
app\main.py:16: async def app_lifespan(app: FastAPI) -> AsyncGenerator[None, None]:

app\core\config.py:9: class Settings(BaseSettings):

app\core\config.py:33: class Config:

app\core\config.py:38: def get_settings() -> Settings:

app\core\db.py:54: async def create_db_pool(dsn: str, min_size: int, max_size: int) -> asyncpg.Pool:

app\core\db.py:69: async def close_db_pool(pool: asyncpg.Pool):

app\core\db.py:82: async def lifespan(app: FastAPI) -> AsyncGenerator[None, None]:

app\core\db.py:127: def _get_pool_from_state(request: Request, pool_name: str) -> asyncpg.Pool:

app\core\db.py:143: def get_db_pool(request: Request) -> asyncpg.Pool:

app\core\db.py:147: def get_db_pool_db2(request: Request) -> asyncpg.Pool:

app\core\db.py:151: def get_db_pool_db3(request: Request) -> asyncpg.Pool:

app\core\db.py:158: async def fetch_single_json(conn: asyncpg.Connection, query: str, *values) -> Optional[Dict[str, Any]]:

app\core\security.py:18: def hash_password(password: str) -> str:

app\core\security.py:21: def verify_password(plain_password: str, hashed_password: str) -> bool:

app\core\security.py:26: def decode_token(token: str, secret_key: str, token_type: str = "access") -> dict:

app\core\security.py:41: def generate_reset_token(email: str) -> str:

app\core\security.py:44: def verify_reset_token(token: str, max_age: int = 3600) -> str:

app\core\security.py:50: def encode_uid(uid: int) -> str:

app\core\security.py:53: def decode_uid(uidb64: str) -> int:

app\crud\client.py:7: async def get_all_clients(db: Pool):

app\crud\client.py:11: async def get_client_by_id(db: Pool, client_id: int):

app\crud\client.py:14: async def get_client_by_user_id(db: asyncpg.Pool, user_id: int) -> Optional[Dict[str, Any]]:

app\crud\client.py:22: async def create_client_user(conn, email, username, hashed_password):

app\crud\client.py:30: async def create_client(db: Pool, data: dict):

app\crud\client.py:48: async def update_client(db: Pool, client_id: int, data: dict):

app\crud\client.py:74: async def delete_client(db: Pool, client_id: int):
```

app\crud\client.py:84: async def change_client_status(db: Pool, client_id: int, new_status: str):

app\crud\client.py:90: async def check_email_exists(db: Pool, email: str):

app\crud\court_wise_data_extractor.py:10: async def fetch_employee_by_department(conn, unique_no: str):

app\crud\court_wise_data_extractor.py:64: async def high_court_fetch(conn, unique_no: str):

app\crud\court_wise_data_extractor.py:113: async def supreme_court_fetch(conn, unique_no: str):

app\crud\court_wise_data_extractor.py:139: async def fetch_state_district_list(conn: asyncpg.Connection) -> List[Dict[str, List[str]]]:

app\crud\credit.py:6: async def update_client_credits(db: Connection, client_id: int, new_balance: int):

app\crud\credit.py:13: async def insert_credit_transaction(

app\crud\credit.py:36: async def get_search_log(db: Connection, search_id: int, client_id: int):

app\crud\search.py:5: async def create_search_log(conn: asyncpg.Connection, data: Dict) -> asyncpg.Record:

app\crud\search.py:22: async def deduct_credits(conn, client_id: int, amount: int, description: str):

app\crud\search.py:37: async def get_search_logs(

app\crud\search.py:120: async def get_client_search_logs(

app\crud\user.py:3: async def get_user_by_id(conn: Connection, user_id: int):

app\crud\user.py:12: async def update_admin_profile(conn: Connection, user_id: int, firstname: str = None, lastname: str = None):

app\crud\user.py:25: async def get_user_by_email(conn: Connection, email: str):

app\crud\user.py:34: async def update_user_password(conn: Connection, user_id: int, hashed_password: str):

app\routes\admin\credit.py:17: async def add_credits(

app\routes\admin\credit.py:59: async def deduct_credits(

app\routes\admin\credit.py:104: async def refund_credits(

app\routes\admin\user.py:16: async def create_superuser(

app\routes\admin\user.py:65: async def get_admin_me(conn=Depends(get_db_pool), current_admin=Depends(get_current_admin)):

app\routes\admin\user.py:73: async def update_admin_me(

app\routes\auth\dependencies.py:12: async def get_current_admin(

app\routes\auth\dependencies.py:31: async def get_current_user(

app\routes\auth\dependencies.py:51: async def get_current_client(

```
app\routes\auth\login.py:13: async def login(request: LoginRequest, db: asyncpg.Pool = Depends(get_db_pool)):

app\routes\auth\login.py:37: async def refresh_token(request: RefreshTokenRequest):

app\routes\auth\password_reset.py:19: async def request_password_reset(data: PasswordResetRequest, conn=Depends(get_db_pool)):

app\routes\auth\password_reset.py:31: async def confirm_password_reset(uidb64: str, token: str, data: PasswordResetConfirmRequest):

app\routes\auth\tokens.py:5: def create_token(data: dict, expires_delta: timedelta, secret_key: str) -> str:

app\routes\auth\tokens.py:11: def create_access_token(data: dict):

app\routes\auth\tokens.py:18: def create_refresh_token(data: dict):

app\routes\client\client.py:15: async def get_current_client_data(client_user=Depends(get_current_client), db: asyncpg.Pool = Depends(get_db_pool)):

app\routes\client\client_admin.py:17: async def list_clients(db: asyncpg.Pool = Depends(get_db_pool), admin=Depends(get_current_client)):

app\routes\client\client_admin.py:47: async def create_client(client_data: dict, db: asyncpg.Pool = Depends(get_db_pool), admin=Depends(get_current_client)):

app\routes\client\client_admin.py:97: async def retrieve_client(client_id: int, db: asyncpg.Pool = Depends(get_db_pool), admin=Depends(get_current_client)):

app\routes\client\client_admin.py:121: async def update_client(client_id: int, client_data: dict, db: asyncpg.Pool = Depends(get_db_pool), admin=Depends(get_current_client)):

app\routes\client\client_admin.py:155: async def delete_client(client_id: int, db: asyncpg.Pool = Depends(get_db_pool), admin=Depends(get_current_client)):

app\routes\client\client_admin.py:162: async def change_status(client_id: int, status_param: str = Query(..., regex="^(activated|deactivated)$"), db: asyncpg.Pool = Depends(get_db_pool), admin=Depends(get_current_client)):

app\routes\common\common.py:13: async def get_states_and_districts(request: Request, client_user = Depends(get_current_client)):

app\routes\common\file_download.py:13: class FileDownloadRequest(BaseModel):

app\routes\common\file_download.py:18: def download_file(request: FileDownloadRequest):

app\routes\common\file_download.py:70: def _get_content_type(filename: str) -> str:

app\routes\common\file_download.py:91: def file_iterator(path: str):

app\routes\search\search.py:26: async def perform_search(query: str, db: asyncpg.Pool = Depends(get_db_pool)):

app\routes\search\search.py:131: async def list_search_logs(db: asyncpg.Pool = Depends(get_db_pool)):

app\routes\search\search.py:164: async def list_search_logs(db: asyncpg.Pool = Depends(get_db_pool)):

app\routes\search\search.py:210: async def get_case_data(case_id: int, db: asyncpg.Pool = Depends(get_db_pool)):

app\schemas\auth.py:3: class LoginRequest(BaseModel):

app\schemas\auth.py:7: class TokenResponse(BaseModel):

app\schemas\auth.py:13: class RefreshTokenRequest(BaseModel):
```

app\schemas\client.py:6: class ClientBase(BaseModel):

app\schemas\client.py:18: class ClientUpdate(BaseModel):

app\schemas\client.py:25: class ClientCreatePublic(ClientBase):

app\schemas\client.py:29: class ClientCreate(ClientCreatePublic):

app\schemas\client.py:32: class ClientOut(ClientBase):

app\schemas\client.py:38: class Config:

app\schemas\client.py:42: class StateDistrictOut(BaseModel):

app\schemas\credit.py:5: class CreditTransactionBase(BaseModel):

app\schemas\credit.py:12: class CreditTransactionCreate(CreditTransactionBase):

app\schemas\credit.py:15: class CreditTransactionOut(CreditTransactionBase):

app\schemas\credit.py:19: class Config:

app\schemas\credit.py:23: class CreditOperation(BaseModel):

app\schemas\credit.py:27: class RefundOperation(BaseModel):

app\schemas\password_reset.py:3: class PasswordResetRequest(BaseModel):

app\schemas\password_reset.py:6: class PasswordResetConfirm(BaseModel):

app\schemas\search.py:5: class SearchLogBase(BaseModel):

app\schemas\search.py:14: class SearchLogCreate(SearchLogBase):

app\schemas\search.py:17: class SearchLogOut(SearchLogBase):

app\schemas\search.py:21: class Config:

app\schemas\search.py:26: class SearchRequest(BaseModel):

app\schemas\search.py:68: class SearchLogResponse(BaseModel):

app\schemas\search.py:84: class CaseDataResponse(BaseModel):

app\schemas\user.py:5: class UserBase(BaseModel):

app\schemas\user.py:15: class UserCreate(UserBase):

app\schemas\user.py:18: class UserOut(UserBase):

app\schemas\user.py:23: class Config:

```
app\schemas\user.py:27: class AdminSelfUpdate(BaseModel):

app\schemas\user.py:31: class AdminSelfOut(UserOut):

app\utility\court_fetch_selector.py:16: def get_court_fetch_function(court_type: str):

app\utility\court_fetch_selector.py:20: def get_court_db_dependency(court_type: str):

app\utility\pagination.py:4: def paginate(data: list, total: int, page: int, page_size: int):

app\utility\response.py:11: def record_to_model(record: Record, model: Type[T]) -> T:

app\utility\response.py:14: def standard_response(message: str, status_code=status.HTTP_200_OK, status_text="success"):

app\utility\search.py:11: def pair_names_with_advocates(

app\utility\search.py:29: class SearchUtility:

app\utility\search.py:31: async def generate_unique_search_id(conn) -> str:

app\utility\search.py:42: def capitalize_case_data(data: Dict[str, Any]) -> Dict[str, Any]:

app\utility\search.py:57: def wrap_string_values_with_quotes(query_params: Dict) -> Dict:

app\utility\search.py:73: async def call_external_api(query_params: Dict) -> Tuple[bool, Dict[str, Any]]:

app\utility\search.py:99: def clean_search_data(response_data: Dict[str, Any]) -> Dict[str, Any]:

app\utility\search.py:173: def get_search_name(data: Dict[str, Any]) -> Optional[str]:

app\utility\send_email.py:14: async def send_reset_email(to_email: str, reset_link: str):

app\utility\send_email.py:34: async def send_client_creation_email(to_email: str, client_name: str, password: str):
```