Great—two months is a solid window to bring this project to life, Adarsh! Here's a focused roadmap to help you make steady, strategic progress each week. This keeps the momentum while balancing learning with building:

---

### 🗓️ \*\*2-Month Learning & Building Plan\*\*

#### \*\*Week 1–2: Backend Bootstrapping with Spring Boot\*\*

- Set up your backend project with Spring Boot.

- Learn to build REST APIs, connect to a database (PostgreSQL/MySQL), and manage users.

- Implement \*\*JWT-based authentication\*\* and roles (e.g. voter, admin).

- Build API endpoints for registration, vote submission, and result retrieval.

➡️ \*Deliverable\*: Working backend with user authentication and a few basic endpoints.

---

#### \*\*Week 3–4: Angular Frontend + UI Integration\*\*

- Set up Angular project and understand component architecture.

- Create forms and use Angular Reactive Forms for login & vote casting.

- Connect Angular to backend via `HttpClient`.

- Polish UI with Bootstrap or Angular Material for a sleek experience.

➡️ \*Deliverable\*: User flow from login → candidate list → cast vote.

---

#### \*\*Week 5: Ethereum & Smart Contracts\*\*

- Learn \*\*Solidity\*\* basics and write a smart contract to manage vote recording and tallying.

- Use \*\*Remix IDE\*\* for initial contract testing.

- Move to \*\*Truffle\*\* or \*\*Hardhat\*\* for local development with \*\*Ganache\*\*.

- Add logic for preventing double voting and final vote count transparency.

➡️ \*Deliverable\*: Deployed smart contract on local Ethereum blockchain.

---

#### \*\*Week 6: Blockchain Integration (Frontend + Backend)\*\*

- Use \*\*Ethers.js\*\* (Angular side) or \*\*Web3j\*\* (Java side) to connect to Ethereum.

- Allow users to interact with MetaMask and submit votes securely to the blockchain.

- Ensure your backend verifies eligibility before allowing blockchain interaction.

➡️ \*Deliverable\*: Voting data securely written to blockchain with eligibility check.

---

#### \*\*Week 7: Testing, Edge Cases, and Admin Tools\*\*

- Test edge cases: duplicate voting, unauthorized access, incorrect input.

- Build simple admin dashboard to view vote stats (read from smart contract).

- Harden security with rate-limiting, CSRF protection, CORS config, etc.

➡️ \*Deliverable\*: Secure, reliable voting system with auditability.

---

#### \*\*Week 8: Polish, Document, and Showcase\*\*

- Prepare \*\*project documentation\*\*: architecture, tech stack, deployment guide.

- Host backend (Spring app) on Render/Heroku, frontend (Angular) on Netlify/Vercel.

- Optional: deploy smart contract to Ethereum testnet (e.g. Goerli via Infura).

- Create a short demo video or presentation deck for internship showcases.

➡️ \*Final Output\*: Fully functional project + deployment + pitch-ready presentation.

---

Would you like me to generate a basic Spring Boot backend architecture or a draft Solidity smart contract to jumpstart Week 1 and Week 5? We can tackle it block by block.