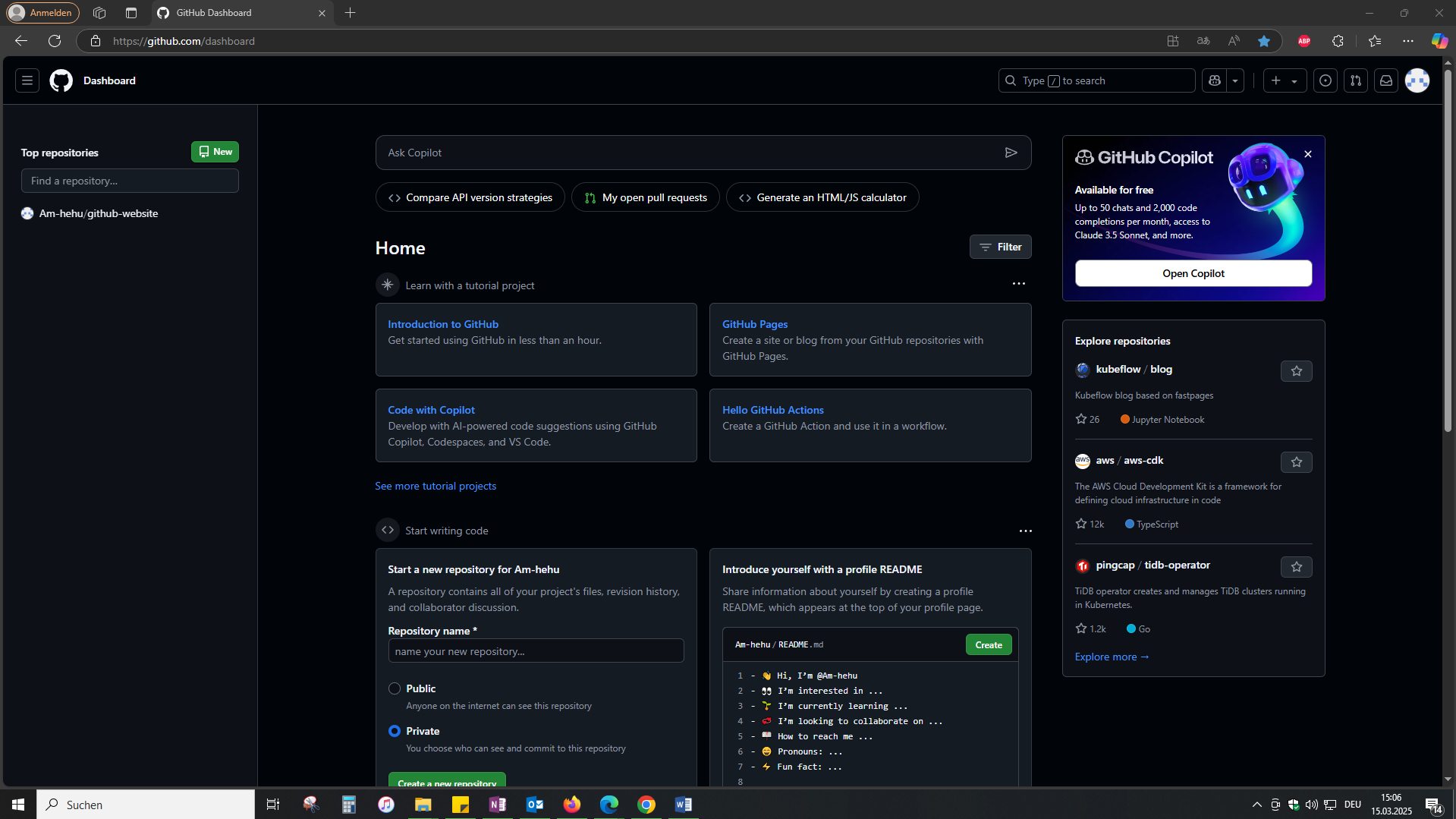
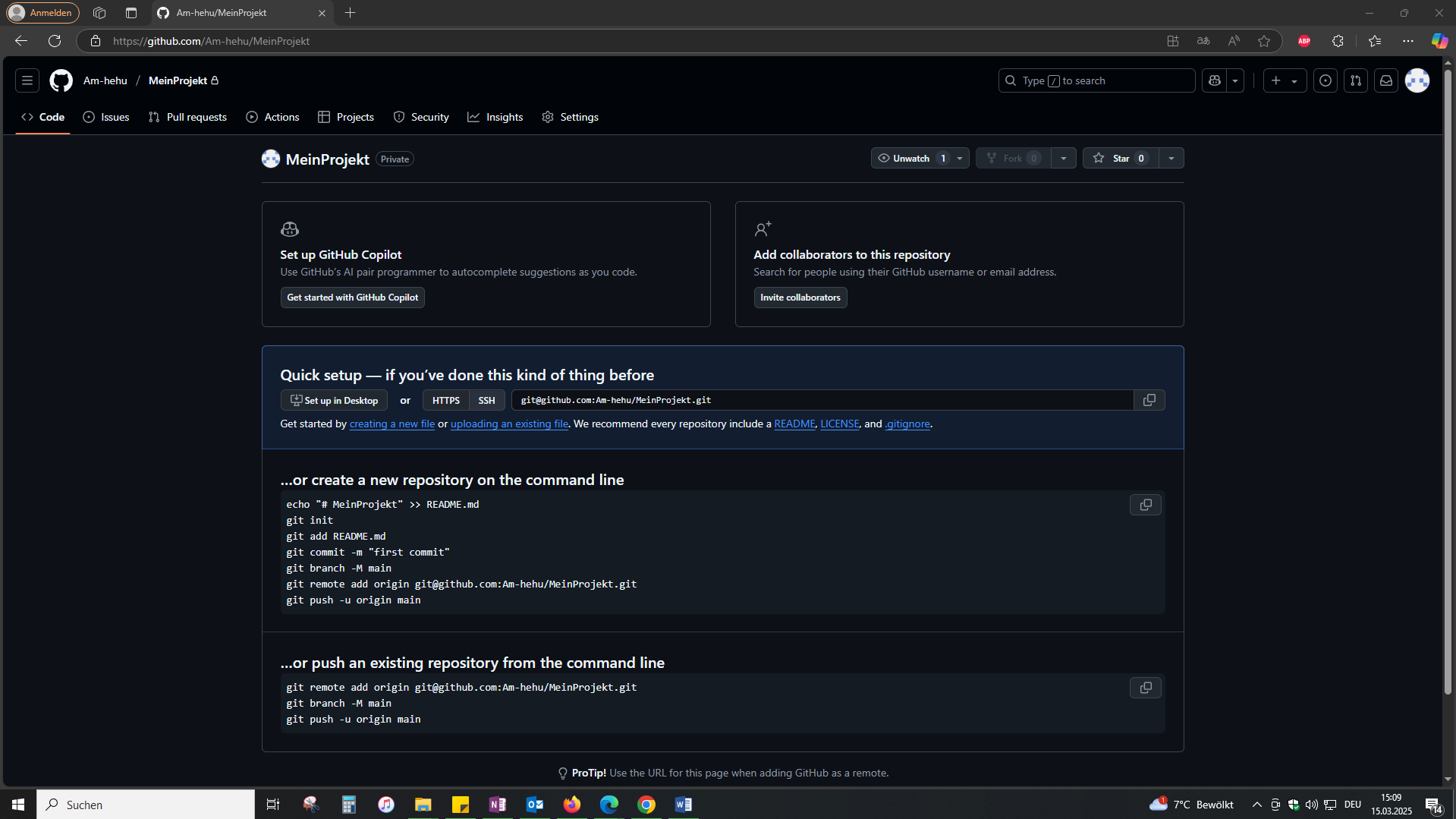
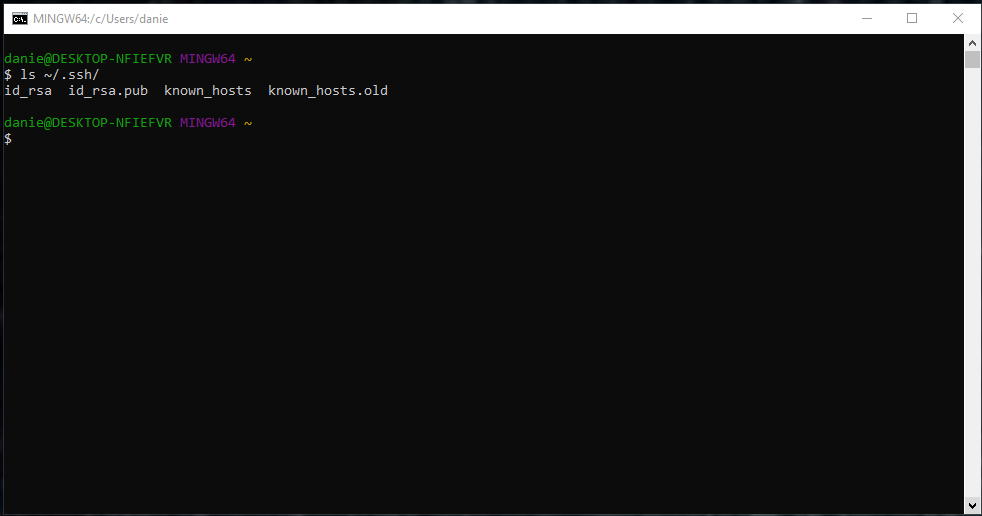
Protokoll

Aufgabe a)

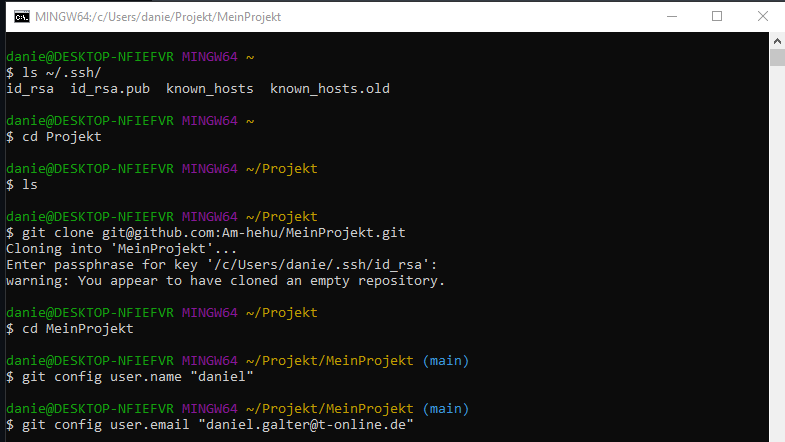
Bild 1: In GitHub angemeldet

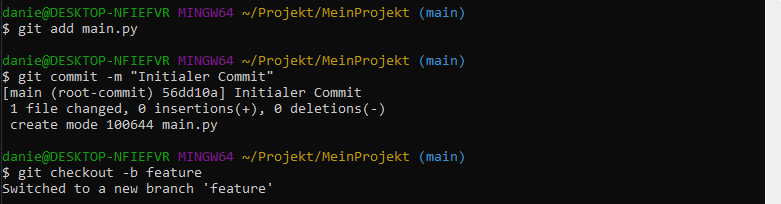
Bild 2: Neues Repository „MeinProjekt“ erstellt mit SSH-URL [git@github.com:Am-hehu/MeinProjekt.git](mailto:git@github.com:Am-hehu/MeinProjekt.git)

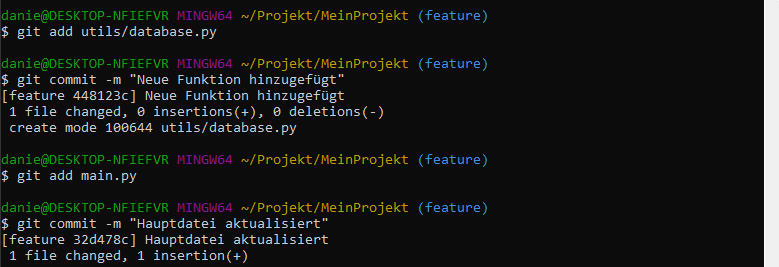
Aufgabe b)

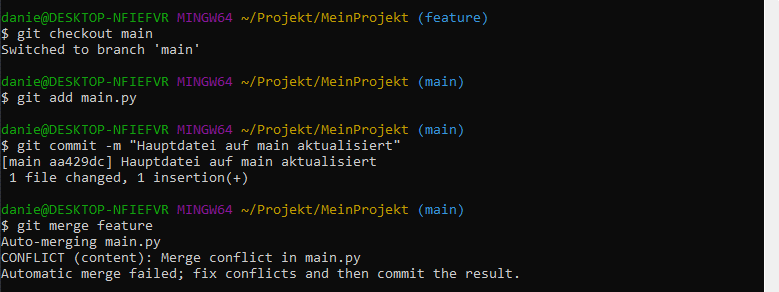
Bild 3: Überprüfung ob bereits ein SSH-Schlüssel vorliegt ist positiv

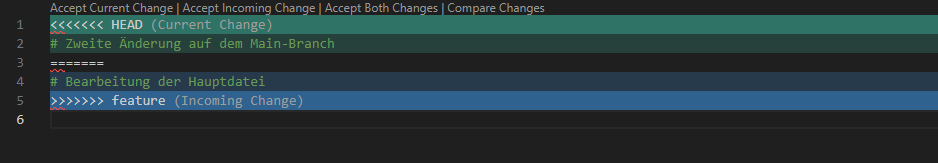
Aufgabe c)

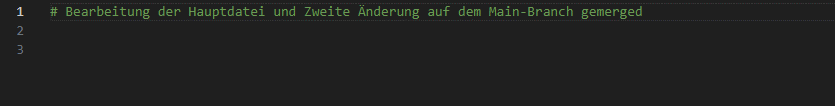
Bild 4: GitHub-Repository geklont und Git mit Namen und Email konfiguriert

Bild 5: Datei „main.py“ hinzugefügt, initialer Commit erstellt und neuen Branch „feature“ erstellt

Bild 6: Datei „Database.py“ hinzugefügt, Commit erstellt, Datei „main.py“ bearbeitet und neuen Commit erstellt

Bild 7: Zum Main-Branch gewechselt, Datei „main.py“ bearbeitet und ein Commit erstellt. Der Versuch die beiden Branchs zu mergen führt zu einem Merge-Konflikt.

Bild 8: In der Merge-Datei sind beide Änderungen aus den jeweiligen Branchs enthalten

Bild 9: Konflikt manuell gelöst durch löschen der Fehlermarkierung und kombinieren der beiden Versionen. Danach die gemergte Datei in den Staging-Bereich mit „$ git add main.py“

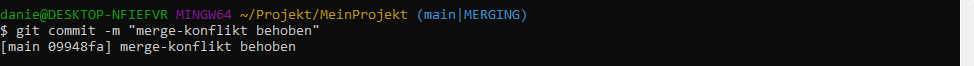
Bild 10: Abschließend die gemergte Datei in neuem Commit speichern

Bild 11: lokales Repository „MeinProjekt“ an mein GitHub-Konto gepusht

Jetzt speichere ich dieses Protokoll im Ordner MeinProjekt, mache ein neues Commit und pushe die neuste Version auf GitHub.